INNOMATICS
RESEARCH LABS

INNOVATION. AUTO**MAT**ION. ANALY**TICS**

PROJECT ON

Code Refactoring and Bug Fixing

Yuvasai Uppala

6th March 2024

About me

I hold a Bachelor of Science (BSc) degree in Chemistry. Despite my non-technical background, my passion for understanding patterns and uncovering insights led me to explore data-driven fields. I embarked on a journey into the fascinating world of data analysis and data science and realized that it offer a unique blend of scientific inquiry and practical problem-solving.

I believe that data-driven decisions can transform businesses, research, and society. By mastering data analysis techniques, I aim to contribute to informed decision-making and drive positive change.

I have a hands on project experience in using MySQL, Excel in different projects across different domains like banking, social media, automobile, human resources and Entertainment industry.

LinkedIn profile : LinkedIn Profile

1. About the Project

A team of enthusiastic data scientists embarked on   mission to develop a Note Taking Application using Python, Flask, and HTML. However, their lack of experience in backend development has led to challenges in making the application fully functional. Recognizing your proficiency in backend development, you have been tasked with fixing the broken code and ensuring the application works seamlessly.

2. Objective/ Task

Refactor the existing codebase and ensure the proper functioning of the Note Taking Application. Document all identified bugs during the debugging process. Remember, the task is not about recreating the app from scratch. Your goal is to fix the already existing codebase and make the application work as intended.

```python
app.py  1 ●        <> home.html

day_0 >  app.py > ...
1    from flask import Flask, render_template, request
2
3    app = Flask(__name__)
4
5    notes = []
6    @app.route('/', methods=["POST"])
7    def index():
8        note = request.args.get("note")
9        notes.append(note)
10       return render_template("home.html", notes=notes)
11
12
13   if __name__ == '__main__':
14       app.run(debug=True)
```

Bug-1:
POST is specified as request method in the route definition, but when you try to access request arguments it only accepts GET requests.

Fix:
Changing route method to accept GET requests or modify your form to submit data using POST method.

Bug-2:
Trying to access form data using request.args.get(), but for a POST request, use request.form.get()

Fix:
Change the request.args.get() to request.form.get() to correctly retrieve form data

```
app.py  1 ●        <> home.html ✕

day_0 > templates > <> home.html > ❖ html
  1    <!DOCTYPE html>
  2    <html lang="en">
  3    <head>
  4        <meta charset="UTF-8">
  5        <meta http-equiv="X-UA-Compatible" content="IE=edge">
  6        <meta name="viewport" content="width=device-width, initial-scale=1.0">
  7        <title>Document</title>
  8    </head>
  9    <body>
 10        <form action="">
 11            <input type="text" name="note" placeholder="Enter a note">
 12            <button>Add Note</button>
 13        </form>
 14
 15        <ul>
 16        {% for note in notes%}
 17            <li>{{ note }}</li>
 18        {% endfor %}
 19        </ul>
 20    </body>
 21    </html>
```
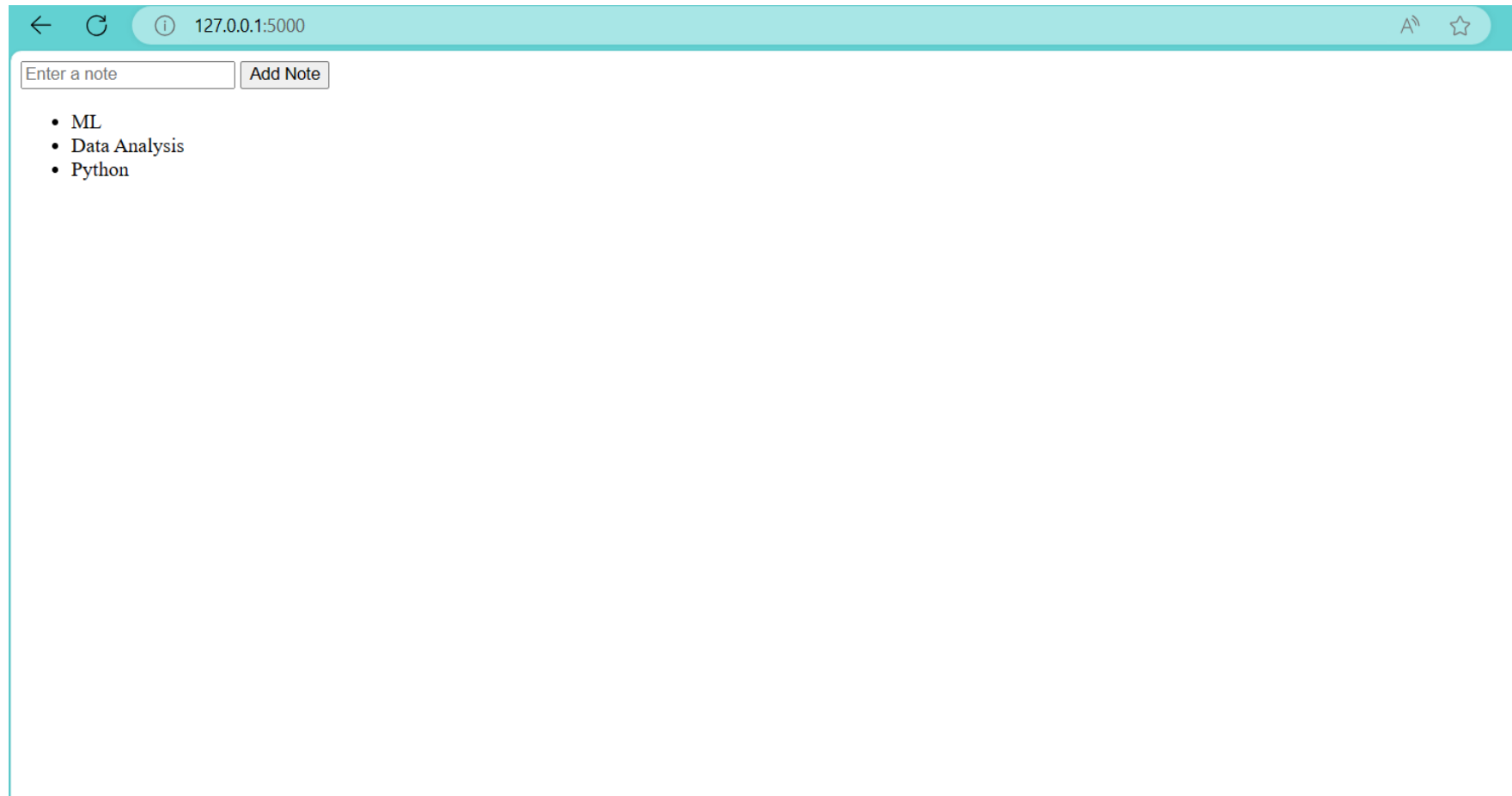
BUG-3:
The button element is missing a type attribute.

Fix:
Adding type = "submit" to the button element.

INNOMATICS
RESEARCH LABS

After debugging the code looks like:

```python
from flask import Flask, render_template, request

app = Flask(__name__)

notes = []
@app.route('/', methods=["GET","POST"])
def index():
    if request.method == "POST":
        note = request.form.get("note")
        notes.append(note)
    return render_template("home.html", notes=notes)


if __name__ == '__main__':
    app.run(debug=True)
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <form action="/", method ="POST">
        <input type="text" name="note" placeholder="Enter a note">
        <button type="submit">Add Note</button>
    </form>

    <ul>
    {% for note in notes%}
        <li>{{ note }}</li>
    {% endfor %}
    </ul>
</body>
</html>
```

This is how it looks like after fixing the bugs.

THANK YOU